A Benders decomposition-based approach for logistics service network design

Simon Belieres, Mike Hewitt, Nicolas Jozefowiez, Frédéric Semet, Tom van Woensel

A Benders decomposition-based approach for logistics service network design

Simon Belieres [1a], Mike Hewitt[b], Nicolas Jozefowiez[c], Frédéric Semet[d], Tom Van Woensel[e]

[a]*CNRS, LAAS, 7 Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France*
[b]*Quinlan School of Business, Loyola University, 16 E. Pearson Ave., IL, Chicago 60611, USA*
[c]*LCOMS EA 7306, Université de Lorraine, Metz 57000, France*
[d]*Université de Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL, F-59000 Lille, France*
[e]*Eindhoven University of Technology, School of Industrial Engineering, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

Abstract

We propose an exact solution method for a Logistics Service Network Design Problem (LSNDP) inspired by the management of restaurant supply chains. In this problem, a distributor seeks to source and fulfill customer orders of products (fruits, meat, napkins, etc.) through a multi-echelon distribution network consisting of supplier locations, warehouses, and customer locations in a cost-effective manner. As these products are small relative to vehicle capacity, an effective strategy for achieving low transportation costs is consolidation. Specifically, routing products so that vehicles transport multiple products at a time, with each product potentially sourced by a different supplier and destined for a different customer. As instances of this problem of sizes relevant to the operations of an industrial partner are too large for off-the-shelf optimization solvers, we propose a suite of techniques for enhancing a Benders decomposition-based algorithm, including a strengthened master problem, valid inequalities, and a heuristic. Together, these enhancements enable the resulting method to produce provably high-quality solutions to multiple variants of the problem in reasonable run-times.

Keywords: Logistics, Service Network Design, Supply Chain, Benders Decomposition

1. Introduction

In this paper, we propose an exact solution method for a Logistics Service Network Design Problem (LSNDP) inspired by the management of restaurant supply chains. In this problem, a distributor seeks to source and fulfill customer orders of products (fruits, meat, napkins, etc.) through a multi-echelon distribution network consisting of supplier locations, warehouses, and customer locations in a cost-effective manner. The primary goal of the LSNDP is to determine a cost-effective transportation plan. As these products are small relative to vehicle capacity, an effective strategy for achieving low transportation costs is consolidation. Specifically, routing products so that vehicles transport multiple products at a time, with each product potentially sourced by a different supplier and destined for a different customer. Depending on the industrial context, the LSNDP may consider different capacity constraints, such as warehouse storage capacity and/or the limits on the number of vehicle departures from a

[1]Corresponding author: sbeliere@laas.fr

facility during a given period of time. While the problem has received little attention to date, one relevant paper is Dufour et al [13], which focuses on managing logistics in the humanitarian sector. The LSNDP can also be viewed as a variant of the Service Network Design Problem (SNDP) [8, 23], which seeks to determine a plan for transporting shipments through a known network of terminals.

While the LSNDP considered in this paper and the SNDP are similar, they also differ in some fundamental ways. In the LSNDP considered in this paper, products flow from suppliers to customers (albeit through a distribution network). Thus, the LSNDP seeks to design a "forward flow" network. The SNDP, on the other hand, makes no presumptions regarding the direction of product/shipment flows. In a sense, these unidirectional flows make the LSNDP easier to solve than the SNDP, as they imply structure to the network that can be exploited algorithmically. That said, most SNDP models studied in the literature presume that the origin and destination locations for each shipment to be transported is specified *a priori*. In the LSNDP, however, customers request delivery of products that may be manufactured by multiple suppliers, as in the Logistics Network Design Problem (LNDP). As a result, the LSNDP involves a sourcing decision, as it determines the supplier and origin location for each transported product request. In this sense, the LSNDP is harder to solve than the general SNDP as it considers an additional dimension of decision-making.

Many real-world instances of the SNDP involve large numbers of shipments, which in turn leads to instances of the SNDP that are too large for off-the-shelf optimization solvers. Inspired by the operations of an industrial partner, we seek to solve similarly-sized instances of the LSNDP. We propose a Benders decomposition-based solution approach similar in spirit to the *Partial Benders Decomposition* approach proposed in [12] for speeding up the solution of scenario-based stochastic programs. In [12], information is derived from the scenarios used to define the stochastic program and used to strengthen the relaxation, referred to as the *master problem*, solved in the course of executing a Benders-type algorithm. Computational results in [12] indicate that the information added to the relaxation greatly strengthens the bound it yields and increases the rate of convergence of the algorithm as a whole. In this paper, we propose a *Partial Benders decomposition*-type strategy for a deterministic network design problem, similar to the work of Fontaine et al. [14] on a different problem.

Traditional Benders-type methods for solving deterministic network design problems solve a master problem where the need to route shipments/products is relaxed, leaving a relaxation whose solution provides a weak bound on the objective function value of the optimal solution to the original problem. We propose strengthening the master problem with variables and constraints that model the need to route a single product that is an aggregation of the different products requested by customers. We prove the validity of this new master problem and with an extensive computational study illustrate that it yields significantly stronger bounds than the master problem traditionally solved. By examining the structures required in a solution to the LSNDP, we derive valid inequalities with which we further strengthen the master problem. Finally, we complement these techniques for strengthening the dual bound with a heuristic for quickly producing high-quality solutions. The primary motivation for these algorithmic developments is to solve a LSNDP inspired by the logistical considerations of our industrial partner. However, to highlight how they can be generalized, we also adapt them to a variant of the LSNDP that models capacity considerations not faced by our industrial partner.

To summarize, this paper makes the following contributions. First, it introduces a new master problem for Benders decomposition-based methods applied to network design problems, particularly those where instances have many products/shipments. Second, it proposes a set of valid inequalities that leverage the information used to reinforce the master. Third, it proposes a heuristic that derives primal solutions from infeasible subproblems. These algorithmic techniques are developed, and their correctness shown, for an optimization model that recognizes many operational considerations seen in supply chain management. Finally, we analyze the results of an extensive computational study to show that collectively, the techniques yield a method that can produce provably high-quality solutions to instances larger than those that have been solved in the literature. We also perform a detailed analysis of the impact of each technique on the performance of the overall method.

The remainder of the paper is organized as follows. In Section 2, we review relevant literature. In Section 3, we introduce decision-makers and formulation of our Logistics Service Network Design. In Section 4, we present the Benders decomposition-based scheme and detail its acceleration techniques. In Section 5, we present and interpret the results of an extensive computational study of the algorithm performance. Finally, in Section 6, we finish with conclusions and a discussion of future work.

2. Literature review

We first review the literature relevant to our problem. Then, as we propose a Benders decomposition-based algorithm, we review the literature relevant to the application of that algorithmic strategy to problems similar to what we seek to solve.

The problem we study, the LSNDP, contains many features that are seen in other, supply chain optimization-type problems. As already noted, the problem is similar to the Service Network Design Problem, in that it focuses on transportation planning decisions within a terminal network. However, it differs in the direction of desired flows through the network, as the LSNDP focuses on a forward flow network from suppliers to customers while the SNDP considers more general flows. While different variants of the SNDP consider different operational considerations regarding asset/resource/vehicle management [17, 11, 22], the nature our industrial partners' logistics operations does not necessitate modeling such concerns.

Arguably the biggest difference between the problem we solve and the variants of the SNDP considered in the literature [8, 23] is that the LSNDP involves a sourcing decision. Specifically, customers place orders for products which are potentially manufactured by multiple suppliers. As a result, the LSNDP must determine which supplier to use to source each order. This is different from the majority of the literature on the SNDP, which focuses on transporting shipments from given origins to given destinations. This sourcing decision adds an inventory management dimension to the problem, as the LSNDP also determines inventory levels of products at warehouses within the distribution network.

Optimizing the sourcing and fulfillment of orders through a multi-echelon distribution network is also considered in supply chain optimization problems [1] such as the Logistics Network Design Problem (LNDP) [21] and the Supply Chain Network Design Problem (SCNDP) [16]. These problems also aim to determine the flow of materials and/or products through a supply chain, as well as inventory levels at warehouses. However, they primarily focus on strategic decisions such as facility location [5, 3] and do not model transportation costs precisely.

In Table 1, we compare the LSNDP to the LNDP/SCNDP and the SNDP. We report the characteristics of these problems, as well as the decisions they adress. Note that this comparison is not based on all possible variants of each problem, but those variants that are considered in the literature. For example, the SNDP can consider shipments that do not have an *a priori* specified origin. However, we are unaware of such a variant being studied in the literature.

Table 1: Comparison of the LNDP/SCNDP, the SNDP and the LSNDP

	Logistic features		Decisions involved				
Problem	Multi-echelon network	Shipment origin not fixed	Location	Production	Distribution	Inventory	Vehicle utilization
LNDP/SCNDP	X	X	X	X	X	X	-
SNDP	-	-	-	-	X	X	X
LSNDP	X	X	-	-	X	X	X

We next turn our attention to the algorithmic strategy we use to solve our problem, Benders decomposition. In 1962, Jacques Benders proposed a decomposition-based algorithm [2] for solving combinatorial optimization problems. This method divides the computational burden of solving a problem into solving a master problem and solving one or more subproblem(s). Solutions to the master problem prescribe values for a subset of the variables, often referred to as first-stage variables. The subproblems are solved to determine values for the remaining variables based on the values of those first-stage variables determined by solving the master. Information from the solution of subproblems is used to determine whether the solution composed of both first-stage and subproblem variable values is optimal. When it is not, that information is then used to generate constraints to add to the master problem, which is then solved again, and the procedure repeats. When subproblems are linear programs, the approach is guaranteed to converge to an optimal solution.

Benders decomposition has been the basis of an effective solution approach for a wide range of problems [6, 18]. However, algorithms based on the standard Benders decomposition are generally inefficient, and require an excessive amount of time and memory before converging. Instead, an effective Benders decomposition-based algorithm typically requires acceleration techniques [15, 7, 18]. Rahmaniani et al. [18] provides an exhaustive review of such techniques, which include strengthening the master problem with problem-specific valid inequalities, techniques for generating constraints from subproblem information that speed up convergence towards an optimal solution, and changes to the decomposition strategy itself. Regarding network design, Costa [6] reviews Benders decomposition-based algorithms for solving problems from this class. More recently, examples of the effectiveness of Benders as the basis of a solution approach for such problems can be found in [20, 7, 14, 24]. Acceleration techniques for Benders applied to Stochastic Network Design problems can be found in [19].

3. Problem definition and mathematical model

In this section, we first define the problem considered in this paper. We then present a mathematical model of that problem.

3.1. Problem definition

We focus on planning the transportation operations for a logistics company tasked with distributing products from suppliers to customers through a distribution network over a fixed planning horizon. In the context of supplying restaurants that are part of the same chain, a customer corresponds to an individual restaurant. That restaurant could then request for the coming month the delivery of napkins (the product) on each Friday at 9 in the morning. Note that products are packed and transported in pallets of homogeneous size that contain a single type of product, and always in the same quantity. Therefore, in the rest of the article we define a unit of product as a pallet of this product. On the supply side, each product is produced at one or more supplier facilities. Each supplier has a limited product line (e.g. a supplier may specialize in paper products). The number of vehicles that can depart a supplier on a daily basis may be limited due to constraints imposed by their outbound logistics operations. Relatedly, there may be limitations on the total quantity of product a supplier may ship each day. However, we presume that delivery requests are communicated long enough in advance to enable production plans that avoid stockouts.

On the demand side, each customer requests deliveries of quantities of products from the distribution company, and do not specify a supplier in the request. Thus, the distribution company must determine how to source each request. While the customer may request the same product to be delivered multiple times over the course of the planning horizon, the quantity requested need not be the same and each request may be sourced from a different supplier. Relatedly, a customer may request delivery of several products that cannot all be sourced by the same supplier. For example, an indivudal restaurant may request a delivery of meats, vegetables and paper products, which requires shipments from several suppliers. To coordinate with their inbound logistics operations, each customer specifies time windows during which product deliveries can occur. These time windows are periodic, e.g. a customer may request deliveries on Friday mornings from 8 to 10 a.m. Thus, each delivery request also includes a delivery day and time window. Note delivery requests need not be periodic. For the first week of a month, a customer may request delivery of two pallets of napkins on Friday between 8 a.m. and 10 a.m. However, for the second week of a month, that same customer may request delivery of one pallet of napkins and one pallet of bananas on Friday between 8 a.m. and 10 a.m.

The distribution company may transport products directly from a supplier facility to a customer location. However, customer order quantities that are typically small relative to vehicle capacity. As a result, the distribution company may instead transport product through a distribution network that connects supplier facilities with customer locations in order to consolidate orders. Terminals within this distribution network are referred to as *Warehouses* and offer both cross-docking and warehousing of products. However, storing product at a warehouse incurs a per-unit, per-unit-of-time cost. A warehouse may also have limited storage for holding products. Like supplier locations, the number of vehicles that can depart a warehouse on a daily basis may be limited. A vehicle dispatched from a supplier to a warehouse or from a warehouse to another warehouse can transport products intended for different customers. However, for this industrial partner, a vehicle dispatched to a customer can only transport products intended for that customer. We illustrate such a network in Figure 1, wherein S_x indicates a supplier facility, C_x indicates a customer location, and W_x indicates a warehouse within the distribution network.

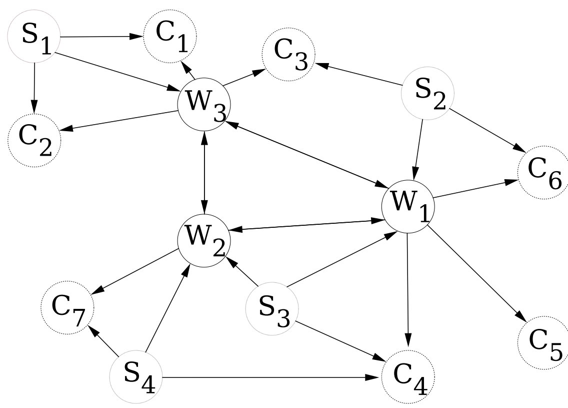

Figure 1: Distribution network

We refer to transportation between warehouses in this network, as well as from a supplier facility or to a customer location, as a *service*. Associated with a service is a departure time from its origin and an arrival time at its destination. While the distribution company plans the execution of services, it relies on a third party carrier for the execution. As this carrier has other customers, the distribution company does not manage the resources needed for transportation (e.g. the distribution company communicates to the carrier needs for point-to-point transportation moves). Relatedly, the distribution company presumes that the carrier's fleet is of sufficient size to satisfy the services it wants executed. In addition to identifying to the carrier which services to execute, the distribution company also specifies a desired capacitiy for each service in fixed units, with each unit of capacity coming at a cost. Ultimately, the distribution company seeks to determine which suppliers satisfy customer requests as well as the services and capacities needed to support those deliveries in order to minimize costs.

3.2. Problem formulation

We model the supply chain with the directed network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, wherein the set $\mathcal{N}$ contains nodes that represent supply locations $\mathcal{S}$, customer locations $\mathcal{C}$, and warehouses $\mathcal{W}$, and the arc set $\mathcal{A}$ contains arcs that represent transportation between such locations. We model the products to be delivered with the set $\mathcal{P}$, with the set supplied by supplier i denoted by $\mathcal{P}^i$. As products are not delivered to suppliers, $\mathcal{A}$ does not contain arcs that model transportation to a supplier. Similarly, as we only consider the delivery of products to customers, $\mathcal{A}$ does not contain arcs that model transportation from a customer. Formally, $\mathcal{A}$ is a subset of $(\mathcal{S} \times \mathcal{W}) \cup (\mathcal{S} \times \mathcal{C}) \cup (\mathcal{W} \times \mathcal{W}) \cup (\mathcal{W} \times \mathcal{C})$. Associated with each arc $a = (i, j) \in \mathcal{A}$, is a travel time $t_{ij} \in \mathbb{N}^*$, a per unit of flow cost $c_{ij} \in \mathbb{R}^{+*}$, a unit of capacity, $\hat{u}$, and a fixed cost per unit of capacity, $f_{ij} \in \mathbb{R}^{+*}$. For the industrial partner that inspired this problem, the unit of capacity models the capacity of one vehicle.

We presume the distribution company seeks to develop a transportation plan for a fixed planning horizon of length $\mathcal{T}$ periods, which is a multiple of the number of days $\mathcal{D}$ in the planning horizon. Thus, there exists a $\Delta \in \mathbb{N}^*$ such that $\mathcal{T} = \mathcal{D} \times \Delta$. As an example, if there are 20 days in the planning horizon, and a period represents half of a day, then $\mathcal{T} = 40$ and $\Delta = 2$. To model the time aspect of the problem, we extend the static network, $\mathcal{G}$, to a time-expanded network $\mathcal{G}_\mathcal{T} = (\mathcal{N}_\mathcal{T}, \mathcal{H}_\mathcal{T} \cup \mathcal{A}_\mathcal{T})$. To construct the graph $\mathcal{G}_\mathcal{T}$, each physical node $i \in \mathcal{N}$ is duplicated $|\mathcal{T}|$ times. As a result, the set $\mathcal{N}_\mathcal{T}$ contains pairs (i, t) for each $i \in \mathcal{N}$ and $t \in \mathcal{T}$. Time-expanded nodes of $\mathcal{N}_\mathcal{T}$ are either time-expanded suppliers $\mathcal{S}_\mathcal{T}$, or time-expanded customers $\mathcal{C}_\mathcal{T}$, or time-expanded warehouses $\mathcal{W}_\mathcal{T}$. Arcs in $\mathcal{H}_\mathcal{T}$ represent storing products at a warehouse. To model this opportunity, for each $i \in \mathcal{W}$ and each $t \in [1, |\mathcal{T}| - 1]$, there is a time-expanded arc $((i, t), (i, t+1))$ in $\mathcal{H}_\mathcal{T}$ with a per-unit-of-flow cost c_{ii}, which represents the per unit, per unit of time storage cost at warehouse i. Note this parameter c_{ii} depends on the length of time modeled by a period in the time-expanded network. Arcs in $\mathcal{A}_\mathcal{T}$ represent transportation between locations as well as departure and arrival times. To construct these, for each $(i, j) \in \mathcal{A}$ and each time $t \in \mathcal{T}$ such that $t + t_{ij} \leq |\mathcal{T}|$, we build a time-expanded arc $((i, t), (j, t + t_{ij}))$. Thus, a transportation arc $((i, t), (j, t + t_{ij}))$ in $\mathcal{A}_\mathcal{T}$ models transporting goods from i to j, leaving at time t and arriving at time $t + t_{ij}$. We note that before creating the network, $\mathcal{G}_\mathcal{T}$, it may be necessary to modify the values t_{ij} to ensure that arcs $(i, j) \in \mathcal{A}$ can be mapped to arcs of the form $((i, t), (j, t + t_{ij}))$. Figures 2 and 3 illustrates how the time dimension of the problem is modelled.

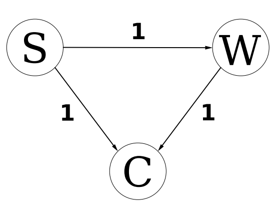

Figure 2: Static network $\mathcal{G}$

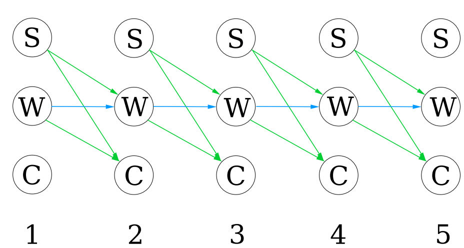

Figure 3: Corresponding time-expanded network $\mathcal{G}_\mathcal{T}$, with $|\mathcal{T}| = 5$

Figure 2 represents a static network with one supplier, one warehouse, one customer and three transportation arcs with a transit time of one unit. Figure 3 illustrates the time-expanded version of that static network, considering a planning horizon of 5 units such that each node of $\mathcal{G}$ is duplicated 5 times in $\mathcal{G}_{\mathcal{T}}$. In Figure 3, transportation and holding arcs are colored in green and blue, respectively.

To model capacity constraints that span multiple periods (e.g. a daily limit on the number of vehicle departures from a warehouse and periods that represent half-days), we link each $t \in \mathcal{T}$ with its corresponding day $d(= \lceil \frac{t}{\Delta} \rceil)$ in the planning horizon. To ease the reading of the paper, we denote that correspondance as $t \in d$. Other model parameters include d^p_{ct}, which is the amount of product $p \in \mathcal{P}$ requested by customer $c \in \mathcal{C}$ to be delivered at time $t \in [1, |\mathcal{T}|]$. Each warehouse $i \in \mathcal{W}$ has a storage capacity: $wlim_i$. The daily supply capacity of a supplier $i \in \mathcal{S}$ is $slim_i$. Finally, the maximum number of vehicles that can be dispatched from a supplier or a warehouse $i \in \mathcal{S} \cup \mathcal{W}$ on each day is $ylim_i$.

Thus, we next formulate the Logistics Service Network Design problem defined over a time-expanded network $\mathcal{G}_{\mathcal{T}}$ (LSND($\mathcal{G}_{\mathcal{T}}$)). The integer variable, $y^{tt'}_{ij}$, represents the number of vehicles dispatched on transportation arc $((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}$. The continuous variable, $x^{ptt'}_{ij}$, represents the quantity of product p that flows along the arc $((i,t),(j,t')) \in \mathcal{H}_{\mathcal{T}} \cup \mathcal{A}_{\mathcal{T}}$ (note as this includes holding arcs, it may be that $i = j$). Also, if $p \notin \mathcal{P}^i$ (i.e. supplier i does not supply product p), then the continuous variables $x^{ptt'}_{ij}$ are not defined for all arcs $((i,t),(j,t'))$. Formally, the **LSND**($\mathcal{G}_{\mathcal{T}}$) seeks to

$$\text{minimize } z(\mathcal{G}_{\mathcal{T}}) = \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} f_{ij} y^{tt'}_{ij} + \sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} \sum_{p \in \mathcal{P}} c_{ij} x^{ptt'}_{ij} + \sum_{((i,t),(i,t+1)) \in \mathcal{H}_{\mathcal{T}}} \sum_{p \in \mathcal{P}} c_{ii} x^{ptt+1}_{ii} \tag{1}$$

Under the following constraints :

$$\sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \cup \mathcal{H}_{\mathcal{T}}} x^{ptt'}_{ij} - \sum_{((j,t'),(l,t'')) \in \mathcal{A}_{\mathcal{T}} \cup \mathcal{H}_{\mathcal{T}}} x^{pt't''}_{jl} = 0, \quad \forall (j,t') \in \mathcal{W}_{\mathcal{T}}, \forall p \in \mathcal{P} \tag{2}$$

$$\sum_{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}} x^{ptt'}_{ij} \geq d^p_{jt'}, \quad \forall (j,t') \in \mathcal{C}_{\mathcal{T}}, \forall p \in \mathcal{P} \tag{3}$$

$$\sum_{p \in \mathcal{P}} x^{ptt'}_{ij} \leq \hat{u} y^{tt'}_{ij}, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \tag{4}$$

$$\sum_{p \in \mathcal{P}} x^{ptt+1}_{ii} \leq wlim_i, \quad \forall ((i,t),(i,t+1)) \in \mathcal{H}_{\mathcal{T}} \tag{5}$$

$$\sum_{\substack{((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \\ t \in d}} \sum_{p \in \mathcal{P}} x^{ptt'}_{ij} \leq slim_i, \quad \forall i \in \mathcal{S}, \forall d \in [1, |\mathcal{D}|] \tag{6}$$

$$\sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}\\ t\in d}} y_{ij}^{tt'} \leq ylim_i, \qquad \forall i \in \mathcal{S}\cup\mathcal{W}, \forall d \in [1, |\mathcal{D}|] \tag{7}$$

$$x_{ij}^{ptt'} \in \mathbb{R}^+, \qquad \forall((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \cup \mathcal{H}_{\mathcal{T}},\ \forall p \in \mathcal{P}^i \tag{8}$$

$$y_{ij}^{tt'} \in \mathbb{N}^+, \qquad \forall((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \tag{9}$$

Objective (1) minimizes the sum of fixed costs on transportation arcs (first term), variable costs on transportation arcs (second term), and variable costs on holding arcs (third term), i.e. holding costs. Constraints (2) enforce flow conservation at each warehouse. Returning to Figure 3, flow conservation constraints associated with node $(w, 2)$ are: $x_{sw}^{p12} + x_{ww}^{p12} = x_{wc}^{p23} + x_{ww}^{p23}, \forall p \in \mathcal{P}$. Constraints (3) impose the respect of each customer demands. Constraints (4) ensure sufficient vehicle capacity is dispatched to transport products. Constraints (5) limit the total amount of product stored by each warehouse. Constraints (6) impose the respect of each supplier's daily supply capacity. Constraints (7) ensure that the daily number of vehicles that can be dispatched from each supplier or warehouse is respected. Constraints (8) and (9) define the variable domains.

4. An enhanced Benders decomposition-based strategy

In this section, we propose an algorithmic strategy for solving LSND($\mathcal{G}_{\mathcal{T}}$). Our method is a Benders decomposition-based scheme based on Partial Benders decomposition [10, 12]. It is enhanced with both valid inequalities (Section 4.2) for strengthening the relaxation solved by the Benders decomposition-based scheme and a heuristic (Section 4.3) to reduce the time needed to find high-quality solutions. We next discuss the scheme in detail.

4.1. Partial Benders decomposition

Benders decomposition is a solution strategy for large mixed-integer linear problems that decomposes a problem into a master problem and a set of subproblems. As we consider a single subproblem in our method, we describe the method in that context. The master problem is a relaxation of the original problem that considers a subset of the variables in the original problem and an estimate of the optimal objective function value of the subproblem. Solving the master problem yields a dual bound on the optimal objective function value of the original problem and variable values that are used to formulate the subproblem that determine values for the remaining variables. When the subproblem is feasible, a feasible solution to the original problem can be constructed. This feasible solution yields a primal bound on the optimal objective function value of the original problem. When the objective function value of the subproblem does not agree with the estimate in the master problem, a Benders cut known as an *Optimality cut* is generated. This type of cut is typically generated from an extreme point of the dual polyhedron associated with the subproblem. When the subproblem is not feasible, a Benders cut known as a *Feasibility cut* is generated. This type of cut is typically generated from an extreme ray of the dual polyhedron associated with the subproblem. Generated cuts are added to the master problem, which is then re-solved. The process repeats until the primal and dual bounds are within some pre-defined optimality tolerance, ϵ, or no Benders cuts are generated.

For the LSND($\mathcal{G}_{\mathcal{T}}$), the standard Benders decomposition yields a master problem that allocates trucks on transportation arcs and a subproblem that routes product flows using the capacity allocated by the master. With Ω and Γ representing the extreme rays and extreme points of the subproblem dual polyhedron, the master problem, **CMP**, is formulated as follows:

$$\min \sum_{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} f_{ij} y_{ij}^{tt'} + z \tag{10}$$

$$\sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}} \\ t\in d}} y_{ij}^{tt'} \leq ylim_i, \qquad \forall i \in \mathcal{S}\cup\mathcal{W}, \forall d \in [1, |\mathcal{D}|] \tag{7}$$

$$0 \geq \sum_{(c,t)\in\mathcal{C}_{\mathcal{T}}} \sum_{p\in\mathcal{P}} d_{ct}^{p}\rho_{ct}^{p} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_{\mathcal{T}}} wlim_i \rho_{ii}^{tt+1} + \sum_{i\in\mathcal{S}} \sum_{d\in[1,|\mathcal{D}|]} slim_i \rho_i^{d} - \sum_{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} \hat{u}\rho_{ij}^{tt'} y_{ij}^{tt'}, \quad \forall \rho \in \Omega \tag{11}$$

$$z \geq \sum_{(c,t)\in\mathcal{C}_{\mathcal{T}}} \sum_{p\in\mathcal{P}} d_{ct}^{p}\pi_{ct}^{p} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_{\mathcal{T}}} wlim_i \pi_{ii}^{tt+1} + \sum_{i\in\mathcal{S}} \sum_{d\in[1,|\mathcal{D}|]} slim_i \pi_i^{d} - \sum_{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} \hat{u}\pi_{ij}^{tt'} y_{ij}^{tt'}, \quad \forall \pi \in \Gamma \tag{12}$$

$$y_{ij}^{tt'} \in \mathbb{N}^{+}, \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \tag{13}$$

$$z \in \mathbb{R}^{+} \tag{14}$$

The objective function, (10), computes the total vehicle costs and an approximation of the costs associated with routing products. Constraints (7) are considered in the master problem as they only involve y variables. Feasibility constraints (11) and optimality constraints (12) are standard Benders cuts added dynamically after solving the subproblem.

Given an allocation of vehicles $\bar{y}$, the subproblem $\mathbf{SP}(\bar{y})$ is formulated as:

$$\min \sum_{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} \sum_{p\in\mathcal{P}} c_{ij} x_{ij}^{ptt'} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_{\mathcal{T}}} \sum_{p\in\mathcal{P}} c_{ii} x_{ii}^{ptt+1} \tag{15}$$

(2)-(3)-(5)-(6)

$$\sum_{p\in\mathcal{P}} x_{ij}^{ptt'} \leq \hat{u}\bar{y}_{ij}^{tt'}, \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \tag{16}$$

$$x_{ij}^{ptt'} \in \mathbb{R}^{+}, \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}} \cup \mathcal{H}_{\mathcal{T}},\ \forall p \in \mathcal{P}^{i} \tag{17}$$

Given a vehicle allocation $\bar{y}$, the subproblem seeks to satisfy customer requests for products, while minimizing the routing and storage costs incurred while doing so. Therefore, the subproblem has the same flow constraints (2)-(3)-(5)-(6) as the complete program. Constraints (16) ensure that on each transportation arc the total flow cannot exceed the available capacity. It has been recognized that this form of decomposition leads to poor computational performance [18] because the master problem and subproblem are unbundled. In particular, the master problem is unlikely to yield a high-quality solution in the early iterations of the algorithm as it is only constrained by the Benders cuts.

To mitigate these problems, Crainic et al. [10, 12] propose a *Partial Benders Decomposition* technique in the context of solving two-stage stochastic programs. The master problem is strengthened by the addition of information derived from the subproblem(s). For our problem, we add to the master problem variables and constraints related to the routing of a *super-product*, χ, that is derived from aggregating all the products $p \in \mathcal{P}$. The demand at a node $(c, t) \in \mathcal{C}_\mathcal{T}$ for this super-product, which we denote by D^χ_{ct}, is obtained by summing the demands for all products to be delivered to customer c at time t: $D^\chi_{ct} = \sum_{p \in \mathcal{P}} d^p_{ct}$. Relatedly, for each arc $((i, t), (j, t'))$ and product p such that a flow variable $x^{ptt'}_{ij}$ is defined in the LSNDP, a super-product flow variable $x^{\chi tt'}_{ij}$ is defined in our master problem. All suppliers can produce the super-product. This fact induces a loss of information as we cannot restrict suppliers to only ship products they manufacture. Figures 4 and 5 illustrate an example, respectively before and after aggregating the products. In this example, the customer demands a unit of product p_1 and a unit of product p_2. The supplier s_1 (respectively, s_2) can only produce p_1 (respectively, p_2).

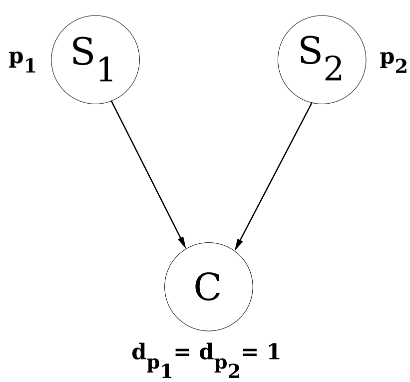

Figure 4: Customer requests one unit of each of two products, each of which supplied by a different supplier.

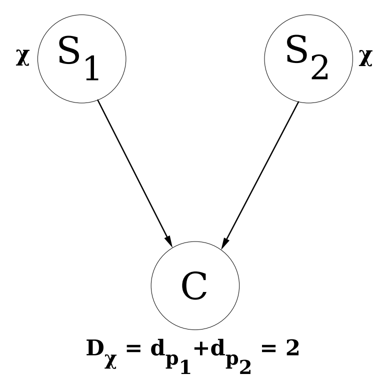

Figure 5: Customer requests two units of the “super-product” that can be supplied by either supplier.

The resulting enhanced master problem (**EMP**) allocates vehicle capacities on transportation arcs in order to satisfy the routing of the super-product:

$$\min \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} f_{ij} y_{ij}^{tt'} + z \tag{18}$$

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{ij}^{\chi tt'} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{jl}^{\chi t't''} = 0, \quad \forall (j,t') \in \mathcal{W}_\mathcal{T} \tag{19}$$

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} x_{ij}^{\chi tt'} \geq D_{jt'}^{\chi}, \quad \forall (j,t') \in \mathcal{C}_\mathcal{T} \tag{20}$$

$$x_{ij}^{\chi tt'} \leq \hat{u} y_{ij}^{tt'}, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \tag{21}$$

$$x_{ii}^{\chi tt+1} \leq wlim_i, \quad \forall ((i,t),(i,t+1)) \in \mathcal{H}_\mathcal{T} \tag{22}$$

$$\sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T} \\ t\in d}} x_{ij}^{\chi tt'} \leq slim_i, \quad \forall i \in \mathcal{S}, \forall d \in [1, |\mathcal{D}|] \tag{23}$$

$$z \geq \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} c_{ij} x_{ij}^{\chi tt'} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_\mathcal{T}} c_{ii} x_{ii}^{\chi tt+1} \tag{24}$$

(7)-(11)-(12)

$$x_{ij}^{\chi tt'} \in \mathbb{R}^+, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T} \tag{25}$$

$$y_{ij}^{tt'} \in \mathbb{N}^+, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \tag{26}$$

$$z \in \mathbb{R}^+ \tag{27}$$

The objective function remains unchanged. Constraints (19) enforce the conservation of super-product flow at each warehouse node. Constraints (20) ensure each customer demand for the super-product is fulfilled. Constraints (21) ensure that enough vehicle capacity is allocated to support the flows of super-product. Constraints (22) limit the total amount of super-product stored by each warehouse. Constraints (23) impose that the flow of super-product shipped per day from a supplier does not exceed its daily supply capacity. Constraints (7) are the same as in the original master problem as they do not involve flow variables. Constraint (24) bounds the flow cost approximation z. Constraints (11) and (12) are the Benders cuts generated dynamically. Constraints (25), (26), and (27) define the decision variables and their domain. It can be shown that this model is a relaxation of LSND($\mathcal{G}_\mathcal{T}$) (see Appendix A for proof). As such, a Benders-based algorithm that solves this master problem will converge to an optimal solution of the LSND($\mathcal{G}_\mathcal{T}$). Next, we describe additional acceleration techniques for improving the performance of our Benders decomposition-based algorithm.

4.2. Valid inequalities

Formulating the **EMP** with an aggregated product leaves a master problem with no knowledge regarding which products each supplier can supply. This loss of information enables the master problem to prescribe vehicle allocations that leave suppliers disconnected from customers, and thus an infeasible subproblem. Thus, to try and prevent such disconnections we next present three valid inequalities with which we strengthen the master problem. The first two seek to ensure that solutions to the master problem induce physical paths from suppliers to customers. The third seeks to ensure that those physical paths reach customers by the times the products they request are to be delivered. We next describe these valid inequalities in detail. The validity of each inequality is proven in Appendix A.

4.2.1. Super-source inequalities

We illustrate this valid inequality with a static network, but it has a natural analog in a time-expanded network. Specifically, Figure 6 illustrates two suppliers, with s_1 manufacturing product p_1 and s_2 manufacturing product p_2. On the demand side, customer c requires one unit of each product. Vehicle capacity is 10. There are transportation arcs (s_1, c) and (s_2, c), but the variable and fixed costs associated with (s_1, c) are less than those with (s_2, c). To formulate the **EMP**, we aggregate products p_1 and p_2 into one super-product χ, which is manufactured by both s_1 and s_2. Customer c's demand of the super-product is obtained by summing the demands of p_1 and p_2, $D_c^{\chi} = \sum_{p \in \mathcal{P}} d_c^p = d_c^{p_1} + d_c^{p_2} = 2$ Given the cost structure in this instance, the optimal solution to the **EMP** is to route 2 units of the super-product from s_1 to c. This solution is illustrated in Figure 7.

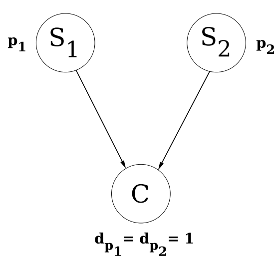

Figure 6: LSND($\mathcal{G}$) Instance

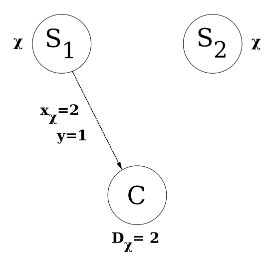

Figure 7: **EMP** optimal solution

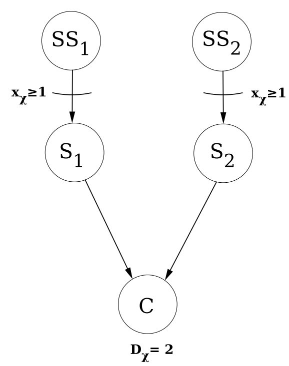

Figure 8: Valid inequalities

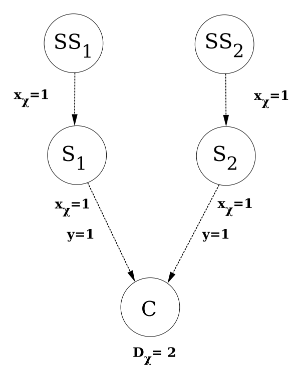

Figure 9: **EMP** optimal solution with inequalities

However, such a solution to the **EMP** will induce an infeasible subproblem as the vehicle allocation does not provide a path from s_2 to c, which is necessary for c to receive product p_2. To avoid such a solution, for each product $p \in \mathcal{P}$ we add to the network what we refer to as a "super-source" ss_p (see figure 8). Then, for each supplier node $s \in \mathcal{S}$ such that $p \in \mathcal{P}^s$, we add to $\mathcal{A}$ the arc (ss_p, s) with zero transit time, linear cost and fixed cost. In addition, we compute the total demand over all customers for each product, $D_p = \sum_{(c,t) \in \mathcal{C}_{\mathcal{T}}} d_{ct}^p$. We then add constraints to **EMP** to ensure that at least D_p units of the super-product is shipped from ss_p and that supplier nodes observe flow conversation with respect to the super-product.

Returning to our example, as the total demand for each of p_1 and p_2 is one unit, the proposed valid inequalities ensure that both ss_1 and ss_2 ship at least one unit of super-product. As the super-sources have outcoming arcs only to the suppliers that manufacture their products, in a solution to the **EMP**, s_1 and s_2 must receive one unit of super-product respectively from ss_1 and ss_2 (see Figure 8). Also, as we enforce flow conservation for s_1 and s_2, any solution to the **EMP** must flow one unit of super-product from s_1 to c and from s_2 to c, meaning the vehicle allocations in the optimal solution to the **EMP** will induce a feasible subproblem (Figure 9).

Formally, we add the following constraints to the **EMP**:

$$\sum_{((ss_p),(j,t))\in\mathcal{A}_{\mathcal{T}}} x^{\chi t}_{ss_p j} \geq D_p, \qquad \forall p \in \mathcal{P} \tag{28}$$

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} x^{\chi tt'}_{ij} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_{\mathcal{T}}} x^{\chi t't''}_{jl} = 0, \qquad \forall (j,t') \in \mathcal{S}_{\mathcal{T}} \tag{29}$$

4.2.2. Direct supply inequalities

Like the previous valid inequality, we illustrate this inequality with a static network, as in Figure 10. We again have that suppliers s_1 and s_2 manufacture products p_1 and p_2, respectively. Now, however, there are two customers, each of which request one unit of both p_1 and p_2. Because each supplier only makes one of the two products requested, the "direct" arcs (s_1, c_1) and (s_2, c_2) cannot fully satisfy those customer demands. Instead, given this network, any feasible solution to the original problem requires that shipments from s_1 and s_2 be transported through warehouse w. To formulate the **EMP**, the products are aggregated, leaving c_1 and c_2 with the following super-product demands: $D^{\chi}_{c_1} = \sum_{p\in\mathcal{P}} d^p_{c_1} = 2$ and $D^{\chi}_{c_2} = \sum_{p\in\mathcal{P}} d^p_{c_2} = 2$. For some cost structures, the optimal solution to the **EMP** will be the solution illustrated in figure 11.

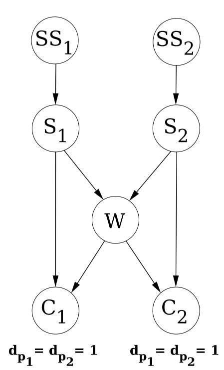

Figure 10: LSND($\mathcal{G}$) instance

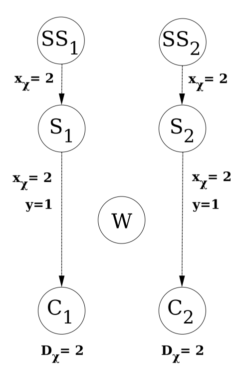

Figure 11: **EMP** optimal solution

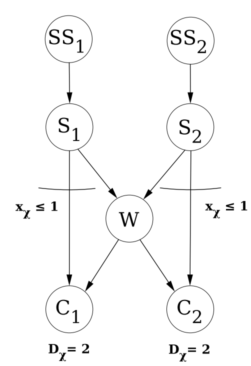

Figure 12: Valid inequalities

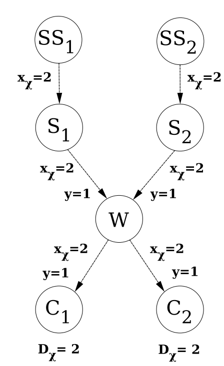

Figure 13: **EMP** optimal solution with inequalities

However, such a solution induces an infeasible subproblem as the vehicle allocations it prescribes do not provide a path from s_1 to c_2 (or from s_2 to c_1). To avoid such an allocation, we use a valid inequality that restricts the flows of super-product on direct arcs. Specifically, given a supplier $s \in \mathcal{S}$ with product set $\mathcal{P}^s$, and a customer $c \in \mathcal{C}$, we compute how much demand c can receive from s: $d^s_c = \sum_{p \in \mathcal{P}^s} d^p_c$. We then restrict the quantity of super-product flow on the direct arc (s, c) to be no greater than d^s_c.

We illustrate these inequalities in Figure 12. As s_1 only manufactures p_1, the flow of super-product on the direct arc (s_1, c_1) cannot exceed $d^{p_1}_{c_1} = 1$. Similarly, the flow of super-product on direct arc (s_2, c_2) cannot exceed $d^{p_2}_{c_2} = 1$. With the inequalities illustrated in Figure 12, an optimal solution to the **EMP** may be the solution illustrated in Figure 13, which induces a feasible subproblem.

In the context of a time-expanded network, given a time-expanded supplier $(s, t) \in \mathcal{S}_\mathcal{T}$ with product set $\mathcal{P}^s$, and a time-expanded customer $(c, t') \in \mathcal{C}_\mathcal{T}$, we denote $d^{st}_{ct'} = \sum_{p \in \mathcal{P}^s} d^p_{ct'}$ as the demand that (c, t') can receive directly from (s, t). Formally, we add the following valid inequality to the **EMP**:

$$x^{\chi tt'}_{sc} \leq d^{st}_{ct'}, \quad \forall ((s,t),(c,t')) \in \mathcal{A}_\mathcal{T},\ (s,t) \in \mathcal{S}_\mathcal{T},\ (c,t') \in \mathcal{C}_\mathcal{T} \tag{30}$$

4.2.3. Time-based Super-source shipment inequalities

Unlike the previous two inequalities, this valid inequality considers the timing of shipment activities. Like the previous two inequalities, we explain this inequality with an example. Specifically, Figure 14 illustrates a time-expanded network associated with the network depicted in Figure 6, wherein the time horizon is 3 days. Customer c's demand is zero for both products at time t_1. However, c requests one unit of each product at times t_2 and t_3. As a result, to formulate the **EMP**, the products are aggregated to yield the following super-product demands:

$$D^\chi_{ct_1} = 0, \ \ D^\chi_{ct_2} = D^\chi_{ct_3} = 2$$

.

A potential optimal solution to the resulting **EMP** is the solution depicted in Figure 15, which does not induce a feasible subproblem as the vehicle allocation does not provide a path from s_2 that arrives at c by day 2, when the delivery of one unit of p_2 is requested. We avoid such allocations in a manner similar to the super-source inequalities described in subsection 4.2.1, but we now consider the timing of shipment activities.

Specifically, for each product $p \in \mathcal{P}$ and each time $t \in \mathcal{T}$, we sum the demands over all customers and obtain a global demand:

$$D^p_t = \sum_{c \in \mathcal{C}} d^p_{ct}, \quad \forall p \in \mathcal{P}, \forall t \in \mathcal{T}$$

Then, for each product $p \in \mathcal{P}$ and each time $t \in \mathcal{T}$, we sum the global demands requested before time t or at time t and obtain a cumulative global demand:

$$\bar{D}^p_t = \sum_{t' \leq t} D^p_{t'}, \ \forall p \in \mathcal{P}, \forall t \in \mathcal{T}$$

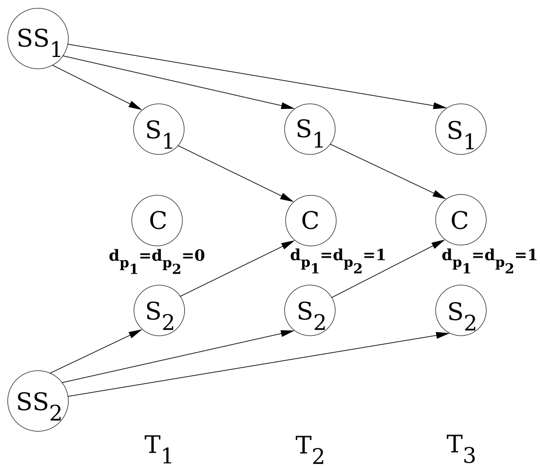

Figure 14: LSND($\mathcal{G}_{\mathcal{T}}$) instance

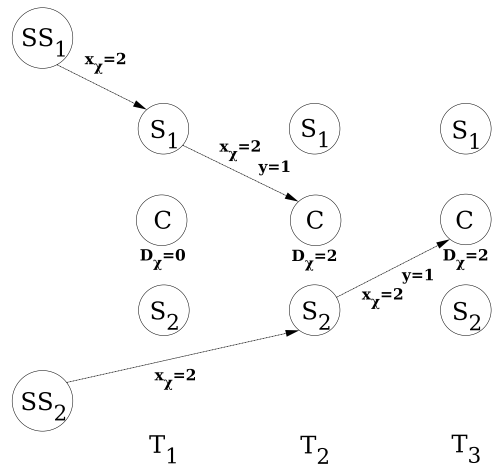

Figure 15: **EMP** optimal solution

For the example considered in Figure 15, the global demands and cumulative global demands are as given in the following tables. The left table corresponds to product p_1, the right table corresponds to product p_2. The first line shows the global demands while the second line shows the cumulative global demands.

t	1	2	3
$D_t^{p_1}$	0	1	1
$\bar{D}_t^{p_1}$	0	1	2

t	1	2	3
$D_t^{p_2}$	0	1	1
$\bar{D}_t^{p_2}$	0	1	2

Given a period t wherein there is an increase in the cumulative global demand for a product (e.g. day 3 for p_1 in our example), we derive a latest time at which that product can be shipped from the corresponding super-source and be delivered on time. To do so, we determine in $\mathcal{G}$ the shortest-path (in terms of time) between each super-source and each customer. Recall that each arc (ss_p, s) from a super-source to a supplier has a null transit time. We denote the length of this shortest path, in terms of time, by $t_{ss_pc}^{min}$. Then, for each super-source, ss_p, we determine the shortest possible delivery time, $t_{ss_p}^{min} = \min_{\forall c \in \mathcal{C}} t_{ss_pc}^{min}$. This duration indicates the smallest transit time between super-source ss_p and a customer for its product. Thus, given a cumulative global demand $\bar{D}_t^p$ such that $\bar{D}_t^p > \bar{D}_{t-1}^p$, if the total amount of product p shipped from super-source ss_p before $t - t_{ss_p}^{min}$ is strictly less than $\bar{D}_t^p$, the demands of product p at time t cannot be satisfied.

In $\mathcal{G}$, the shortest-path duration from ss_2 to c is $t_{ss_2c}^{min} = 1$. As c is the only customer, we have that $t_{ss_2}^{min} = 1$. Thus, for each $t^* \in \mathcal{T}$ such that $\bar{D}_{t^*}^{p_2} > \bar{D}_{t^*-1}^{p_2}$, we must enforce that the flow of super-product from ss_2 to supply nodes (s,t) with $t \leq t^* - t_{ss_2}^{min}$ is at least $\bar{D}_{t^*}^{p_2}$. For example, as $\bar{D}_{t_2}^{p_2} > \bar{D}_{t_1}^{p_2}$ the super-product flow from ss_2 to (S_2, T_1) must be greater or equal to $\bar{D}_{t_2}^{p_2} = 1$, which is not the case in the solution depicted in Figure 15. Similar reasoning can be applied to p_1. We illustrate these valid inequalities in Figure 16 and the resulting optimal solution to the **EMP** in Figure 17, which will induce a feasible subproblem.

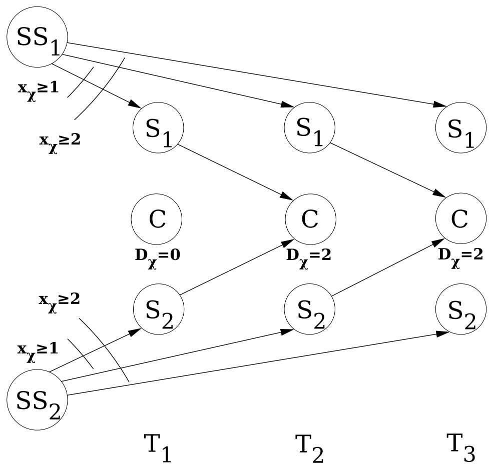

Figure 16: Valid inequalities

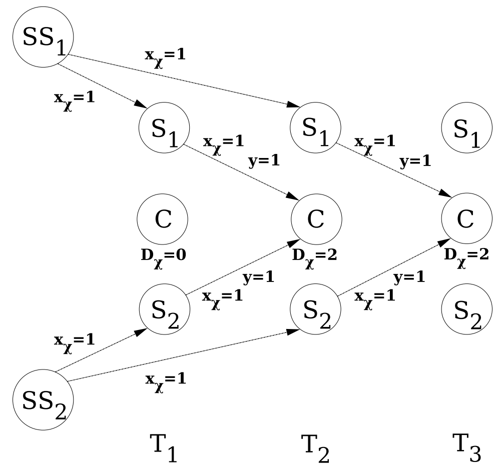

Figure 17: **EMP** optimal solution with inequalities

Formally, for each product $p \in \mathcal{P}$ and each time $t^* \in \mathcal{T}$ such that $\bar{D}^p_{t^*} > \bar{D}^p_{t^*-1}$ - we add the following constraints to **EMP**:

$$\sum_{\substack{((ss_p),(j,t))\in\mathcal{A}_\mathcal{T}\\ t\leq t^*-t^{min}_{ss_p}}} x^{\chi t}_{ss_p j} \geq \bar{D}^p_{t^*}, \quad \forall p \in \mathcal{P},\ \forall t^* \in \mathcal{T},\ \bar{D}^p_{t^*} > \bar{D}^p_{t^*-1} \tag{31}$$

4.3. Slope scaling heuristic

A standard Benders decomposition-based solution method only produces primal solutions when the solution to the master problem induces a feasible subproblem. Thus, to speed up the search for high-quality primal solutions, we propose a heuristic that will derive primal solutions from a vehicle allocation, $\bar{y}$, that induces an infeasible subproblem. In short, we first determine whether we should attempt to repair the vehicle allocation, $\bar{y}$, so that it may yield a feasible subproblem, and then we repair that allocation. We next describe each step in detail. Algorithm 1 provides a high-level description of the procedure.

To determine whether to repair a vehicle allocation, $\bar{y}$, we formulate a subproblem $\mathbf{SP}_s(\bar{y})$ with slack variables to identify how "close" the subproblem is to being feasible given that allocation. The premise being that the closer the subproblem is to being feasible, the more likely a high-quality solution can be derived by making just a few modifications to $\bar{y}$. The subproblem, $\mathbf{SP}_s(\bar{y})$, is formulated as follows:

$$\min \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} \sum_{p\in\mathcal{P}} c_{ij} x^{ptt'}_{ij} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_\mathcal{T}} \sum_{p\in\mathcal{P}} c_{ii} x^{ptt+1}_{ii} + \sum_{(i,t)\in\mathcal{C}_\mathcal{T}} \sum_{p\in\mathcal{P}} c^{prohib} s^p_{it} \tag{32}$$

(2)-(5)-(6)-(16)

Algorithm 1 Build heuristic solution

Require: EMP solution $(\bar{x^{\chi}}, \bar{y})$, threshold $\bar{r}$

if $\text{SP}(\bar{y})$ is infeasible **then**

  Add the corresponding Benders feasibility cut to the master

  Build $\text{SP}_s(\bar{y})$ with slack variables s_{jt}^p for each customer's demand

  Solve $\text{SP}_s(\bar{y})$ to obtain $(\dot{x}, \dot{s})$

  Evaluate the percentage, r, of demand quantities, d_{jt}^p, served with slack variables

  if $r < \bar{r}$ **then**

    Determine initial vehicle allocation $\dot{y}$ from $\dot{x}$

    for Each demand d_{jt}^p served by slack variables in decreasing order **do**

      Route demand d_{jt}^p with a slope-scaling linear program

      Update $(\dot{x}, \dot{y})$

    end for

    if $(\dot{x}, \dot{y})$ has a better objective value than the incumbent and is feasible for the original program **then**

      Update the incumbent

    end if

    Solve $\text{SP}(\dot{y})$ and add the corresponding Benders cut to the master

  end if

end if

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} x_{ij}^{ptt'} + s_{jt'}^{p} \geq d_{jt'}^{p} \qquad \forall (j,t') \in \mathcal{C}_\mathcal{T}, \forall p \in \mathcal{P} \tag{33}$$

$$x_{ij}^{ptt'} \in \mathbb{R}^+, \qquad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, \forall p \in \mathcal{P}^i \tag{34}$$

$$s_{it}^{p} \in \mathbb{R}^+, \qquad \forall (i,t) \in \mathcal{C}_\mathcal{T} \tag{35}$$

This linear program differs from the original subproblem by the extra slack variables, s_{jt}^p, which appear in the objective, and the replacement of constraints (3) with constraints (33). We note that the slack variables guarantee that this suproblem is feasible. The method we propose chooses an objective function coefficient, c^{prohib}, for these slack variables that is high enough that an optimal solution to $\mathbf{SP}_s(\bar{y})$ will only assign positive values to the slack variables when the original subproblem is infeasible. Given an optimal solution $(\dot{x}, \dot{s})$ to $\mathbf{SP}_s(\bar{y})$, we compute the percentage of customer demands that cannot be met with the allocation $\bar{y}$:

$$r = \frac{\sum\limits_{(i,t)\in\mathcal{C}_\mathcal{T}} \sum\limits_{p\in\mathcal{P}} \dot{s}_{it}^{p}}{\sum\limits_{(i,t)\in\mathcal{C}_\mathcal{T}} \sum\limits_{p\in\mathcal{P}} d_{it}^{p}}$$

This measure is our indicator of how "close" the allocation of vehicle capacities, $\bar{y}$, is to inducing a feasible subproblem. We compare this percentage with a threshold, $\bar{r}$, to determine whether we should attempt to repair the solution $\bar{y}$ so that it induces a feasible subproblem, $\mathrm{SP}(\bar{y})$.

Given a vehicle allocation that is to be repaired, the heuristic determines the minimum vehicle allocation needed to route the product flows, $\dot{x}_{ij}^{ptt'}$, specified by the subproblem. Specifically, the heuristic computes $\dot{y}_{ij}^{tt'} = \left\lceil \frac{\sum_{p\in\mathcal{P}} \dot{x}_{ij}^{ptt'}}{\hat{u}} \right\rceil, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$. The heuristic then iterates through customer demands served by slack variables in decreasing order of size, $d_{jt'}^{p}$, and finds a route for each demand via a slope-scaling-type ([9]) procedure that we next describe.

The slope-scaling procedure for demand request $d_{jt'}^{p}$ begins by computing the remaining capacity on each arc given the vehicle allocations, $\dot{y}_{ij}^{tt'}$. It does so by computing $res_{ij}^{tt'} = \hat{u}\dot{y}_{ij}^{tt'} - \sum\limits_{p\in\mathcal{P}} \dot{x}_{ij}^{ptt'}$, $\forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T}$. Then, the procedure determines how many "extra" vehicles are needed on an arc if it is to transport the demand request. Specifically, it calculates for $d_{jt'}^{p}$ the quantity $extra_{ij}^{tt'} = \max\left(0, \left\lceil \frac{d_{jt'}^{p} - res_{ij}^{tt'}}{\hat{u}} \right\rceil\right)$. These quantities are then used to compute the terms, $\widetilde{c}_{ij}^{tt'}$, that linearize the fixed costs associated with allocating additional vehicles to arcs. Specifically, the quantities $\widetilde{c}_{ij}^{tt'} = \frac{c_{ij} d_{jt'}^{p} + f_{ij} extra_{ij}^{tt'}}{d_{jt'}^{p}} = c_{ij} + \frac{f_{ij} extra_{ij}^{tt'}}{d_{jt'}^{p}}$ are computed. In addition, to take account of the storage capacities and the daily supply capacities, the procedure determines the remaining storage capacities, $\dot{x}_{ii}^{tt+1}$, and the actual amount of products shipped per day per supplier, $\dot{x}_{i}^{d}$. More specifically, it computes $\dot{x}_{ii}^{tt+1} = \sum\limits_{p\in\mathcal{P}} \dot{x}_{ii}^{ptt+1}$, $\forall ((i,t),(i,t+1)) \in \mathcal{H}_\mathcal{T}$, and $\dot{x}_{i}^{d} = \sum\limits_{\substack{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T} \\ t\in d}} \sum\limits_{p\in\mathcal{P}} \dot{x}_{ij}^{ptt'}$, $\forall i \in \mathcal{S}, \forall d \in [1, |\mathcal{D}|]$. Based on these terms, we formulate and solve the following linear program for routing $d_{jt'}^{p}$.

$$\min \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} \widetilde{c}_{ij}^{tt'} x_{ij}^{tt'} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_\mathcal{T}} c_{ii}^{tt+1} x_{ii}^{tt+1} \tag{36}$$

subject to

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{ij}^{tt'} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{jl}^{t't''} = 0, \quad \forall (j,t') \in \mathcal{W}_\mathcal{T} \tag{37}$$

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} x_{ij}^{tt'} \geq d_{jt'}^{p}, \quad \forall (j,t') \in \mathcal{C}_\mathcal{T} \tag{38}$$

$$x_{ii}^{tt'} \leq wlim_i - \dot{x}_{ii}^{tt'}, \quad \forall ((i,t),(i,t+1)) \in \mathcal{H}_\mathcal{T} \tag{39}$$

$$\sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T} \\ t\in d}} x_{ij}^{tt'} \leq slim_i - \dot{x}_i^{d}, \quad \forall i \in \mathcal{S}, \forall d \in [1, |\mathcal{D}|] \tag{40}$$

$$x_{ij}^{tt'} \in \mathbb{R}^+, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}., p \in \mathcal{P}^i \tag{41}$$

The objective function of this linear program computes the total approximated routing costs on transportation arcs and storage costs associated with holding arcs. Flow conservation is enforced by (37), while the satisfaction of demand $d_{jt'}^{p}$ is enforced by (38). Constraints (39) ensure that warehouse storage capacities are not exceeded. Constraints (40) ensure that daily supply capacities are not violated while routing the demand request. Given a solution $\ddot{x}$ to this linear program, the heuristic updates the overall solution, $(\dot{x}, \dot{y})$ as follows:

$$\dot{y}_{ij}^{tt'} = \left\lceil \frac{\dot{x}_{ij}^{ptt'} + \ddot{x}_{ij}^{tt'}}{\hat{u}} \right\rceil, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T},$$

$$\dot{x}_{ij}^{ptt'} = \dot{x}_{ij}^{ptt'} + \ddot{x}_{ij}^{tt'}, \quad \forall ((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}.$$

After executing these steps for each demand served by slack variables, we obtain a heuristic solution that respects constraints (2)-(3)-(4)-(5)-(6). Recall that constraints (7) are not modeled in the slope scaling heuristic, if the obtained heuristic solution also respects constraints (7) then it is feasible for the original problem. In that case, if the the current best solution provides a higher bound than the newly-found solution, we update it. In addition, note that we solve subproblem $SP(\dot{y})$ whether the heuristic solution found is feasible or not. This enables us to generate a new Benders cut for the master.

5. Computational study

In this section, we assess the efficiency of our proposed algorithm through a computational study. We first describe the instances used and then how the study was performed. We then analyze results from that study derived from solving two variants of the LSNDP with the proposed algorithm. The first variant is inspired by the operational considerations of our industrial partner. In this model, constraints (6) and (7) are removed, leaving only capacity constraints related to warehouses. While we focus much of our analysis on this case, we also analyze the performance of the algorithm on a second model wherein all presented constraints are in place to study its performance on a more general problem.

5.1. Instances

We tested the algorithm on a set of instances produced by a random generator inspired by the operations of our industrial partner. We next describe this generator, including its key parameters. We first describe how it generates a physical network. One parameter to the generator is the size of the node set. Given that size, the generator randomly generates the graph, $\mathcal{G}$, on a square area of size 100×100. In all instances, 30% of the nodes are labeled as supplier locations and 50% labeled as customer locations. This distinction is made randomly and the percentages are based on the supply chain of our industrial partner. Amongst the remaining nodes, two are labeled as central warehouses whereas the remaining are labeled as regional warehouses. This ratio of central to regional warehouses is also based on the supply chain of our industrial partner.

Regarding transportation arcs, $\mathcal{A}$ contains an arc from each supplier to the nearest central warehouse and from each central warehouse to each regional warehouse. In addition, there is a transportation arc in $\mathcal{A}$ to each customer from its nearest regional warehouse. A second parameter to the generator, α, is a connectivity radius value that is used to determine other arcs in $\mathcal{A}$. Specifically, transportation arcs are added to $\mathcal{A}$ for pairs of locations that are less than α units apart.

The travel times and fixed costs for an arc are set to be proportional to its length. For the travel time, we set a maximum of $\bar{t}_{ij}^{max} = 24h$. We calculate d^{min} and d^{max}, the smallest and largest distances between nodes of $\mathcal{G}$. Then, given two nodes i and j with distance d_{ij}, we set the travel time as: $\bar{t}_{ij} = \bar{t}_{ij}^{max} * \left(\frac{d_{ij} - d^{min}}{d^{max} - d^{min}}\right)$. The truck fixed cost is set to 0.55 per unit of distance: $f_{ij} = 0.55 d_{ij}$. Finally, on each arc we set a flow cost of 0.4 for loading and unloading each pallet of product, yielding $c_{ij} = 0.8$.

The temporal aspect of an instance is determined by two more parameters: (1) D, the number of days in the planning horizon, and, (2) Δ, the time granularity. The time granularity Δ expresses the number of time points per day in the time-expanded graph. For example, if $\Delta = 2$ there are 2 time points per day and each pair of contiguous time points is separated by a time interval of 12 hours. Then, the time horizon for the model is $T = [1, D \times \Delta]$. To express the travel time of an arc in terms of time points, we set $t_{ij} = \lceil \bar{t}_{ij} * \Delta/24 \rceil$, where the original travel time of arc $(i, j) \in \mathcal{G}$ is $\bar{t}_{ij}$.

The last parameter to the generator is the size of the product set, P. Regarding suppliers, each supplier has a 15% chance of manufacturing a product. Regarding customers, each customer has a demand for each product one, two or three times each week with the determination made randomly. These days are chosen randomly. The volume of each demand is randomly chosen in the interval $[0; 5]$. Vehicle capacities are set to 60.

Regarding values of the capacity parameters, discussions with the industrial partner indicate that the maximum capacities ($wlim_i$) of central and regional warehouses are 10,000 and 5,000 pallets, respectively. For each supplier, we randomly chose a daily supply capacity ($slim_i$) from the interval $[200, 300]$. The number of vehicles that can be dispatched from a facility on each day ($ylim_i$) is based on the number of vehicles that can be loaded hourly at that facility. We assume that no more than 4 vehicles can be processed hourly by a supplier or a regional warehouse. We assume that no more than 8 vehicles can be loaded hourly at a central warehouse. As facilities operate 16 hours per day, no more than 64 vehicles can be dispatched daily from a supplier or a regional warehouse. Similarly, no more than 128 vehicles can be dispatched daily from a central warehouse.

For our experiments, we generate instances based on the following values for the other parameters: $|\mathcal{G}| = \{50\}$, $\alpha = \{10, 30\}$, $D = 30$ days, $\Delta = \{2, 3, 4\}$, and $|P| = \{100, 200, 300, 400, 500\}$. Thus, there are 30 possible combinations of parameter values and for each combination we generated 5 instances, leaving 150 instances in total.

5.2. Setup of study

To assess the efficiency of each component of the proposed algorithm, we tested several methods on the instances detailed above. The first method, **SPBD**, is the Partial Benders decomposition-based scheme, wherein the super-product master problem, **EMP**, is used, but the valid inequalities and heuristic are not used. Then, to test the effectiveness of the valid inequalities, the methods **SPBD1**, **SPBD2** and **SPBD3** are Super-Product Benders Decomposition methods supplemented with the a priori cuts described in subsections 4.2.1 (**SPBD1**), 4.2.2 (**SPBD2**), and 4.2.3 (**SPBD3**). The method **SPBD123** employs all three a priori cuts. The method **SPBD123H** adds the proposed heuristic.

We consider three other methods as benchmarks. The first method, **CBD** is the Classic Benders Decomposition, wherein none of the enhancements proposed in this paper are used. The second, **CPLEX**, is the CPLEX implementation of branch-and-cut. The last, **CPLEX-Benders**, is CPLEX's implementation of an automatic Benders decomposition.

All Benders decomposition-based methods are implemented with the callback framework wherein subproblems are solved within the context of the branch-and-bound tree used to solve the master problem. Specifically, whenever an integral solution is found in the tree, the subproblem is solved. The resulting cut is then embedded in every node of the tree, and may cut-off the incumbent. The process terminates once the optimality gap is closed. We initiate every method with a heuristic solution (x_h, y_h) obtained by setting each vehicle variable, $y_{ij}^{tt'}$ to the ceiling of its value in the optimal solution of the linear relaxation of the LSND($\mathcal{G}_\mathcal{T}$). Note that while we implemented versions of the Benders decomposition-based methods that generated pareto-optimal cuts ([15]), doing so did not improve performance.

All algorithms are coded in C++ and executed on an Intel Xeon E5-2695 processor with 16 GB of memory under Linux 16.04. Linear and integer programs were solved using Cplex 12.7. All algorithms are executed with a stopping criteria of a proven optimality gap of 1% or less and a maximum run-time of 1.5 hours. For **SPBD123H**, The threshold parameter, $\bar{r}$, on the percentage of customer demands that can not be fulfilled with the vehicle allocations prescribed by the solution to the master problem for the allocation to be repaired by the heuristic is set to 30%. This value was determined through a set of tuning experiments.

5.3. Results for variant of the LSNDP relevant to industrial partner

Due to its relevance to our industrial partner, we next investigate the performance of our Benders strategy on the first variant of the LSND, which does not include constraints (6) and (7).

5.3.1. Effectiveness of **SPBD123H**

We first benchmark **SPBD123H** against **CBD**, **CPLEX**, and **CPLEX-Benders** by comparing optimality gaps at termination for each method. We note that none of the instances could be solved by any method in the time limit given. We present in Table 2 averages of these gaps over instances with the same number of products $|\mathcal{P}|$. We present more detailed results in Appendix B. Specifically, Table B.1 reports average lower and upper bounds

Table 2: Optimality gaps comparison of **CBD**, **CPLEX-Benders**, **CPLEX** and **SPBD123H**

$\|\mathcal{P}\|$	CBD Opt. gap	CPLEX-Benders Opt. gap	CPLEX Opt. gap	SPBD123H Opt. gap
100	98.48%	77.01%	13.10%	4.89%
200	98.94%	75.67%	40.11%	6.68%
300	99.19%	74.98%	59.24%	6.65%
400	99.33%	73.63%	69.36%	5.64%
500	99.60%	76.89%	76.89%	4.35%

on the optimal objective function values. To give a fuller picture of the relative performance of the methods, we display distributions (in deciles) of the gaps for $|\mathcal{P}| = 300$ in Figure 18. We note that the distributions for other numbers of products are similar.

We observe that **SPBD123H** yields better gaps at termination, on average, than our three benchmarks for every set of instances. We also observe that the performance of **CPLEX** degrades as the number of products increases. On the other hand, **SPBD123H** remains effective for the larger instances. We also observe that the average gap at termination reported by **SPBD123H** decreases as the number of products increases. We will

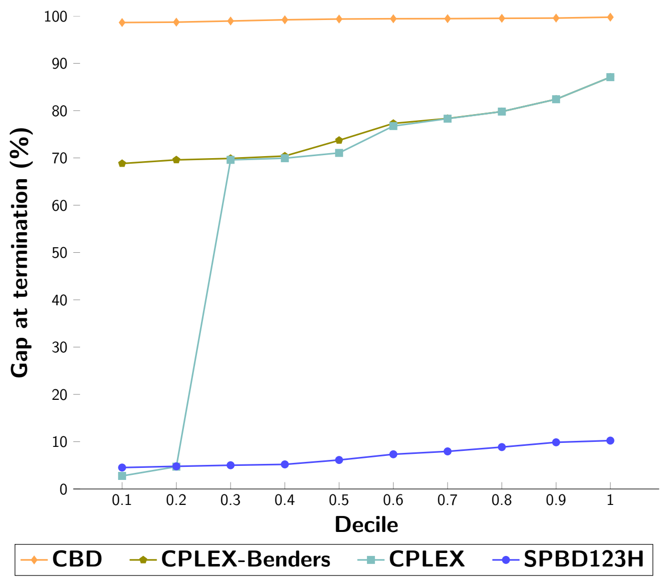

Figure 18: Gap at termination distribution for $|\mathcal{P}| = 300$

analyze why this occurs later. For another perspective, we report in Table 3 the relative gap between the lower and upper bounds produced by **SPBD123H** and the best lower and upper bounds produced by any method.

Table 3: Relative gaps between the LB/UB computed by SPBD123H and the best LB/UB found over all methods

$\|\mathcal{P}\|$	LB relative gap	UB relative gap
100	0.84%	0.00%
200	0.54%	0.18%
300	0.36%	0.22%
400	0.06%	0.00%
500	0.00%	0.00%

We clearly see that the lower optimality gaps yielded when executing **SPBD123H** are because **SPBD123H** nearly always produces the strongest lower bound and highest-quality primal solution. Having established the effectiveness of **SPBD123H**, we next turn our attention to how its features impact its ability to produce a strong lower bound.

5.3.2. Improving the lower bound

We first study the impact of using the super-product master problem, **EMP**, on the lower bound produced at termination. To do so, we report in Table 4 the average lower bound reported by **SPBD** and each of our three benchmarks at termination. We see that **CBD** yields a very weak lower bound, while **CPLEX** and **CPLEX-Benders** produce stronger lower bounds. **SPBD** produces the strongest lower bound, one that is 32.99% greater, on average, in value than the bound produced by the best benchmark (**CPLEX**). Thus, we conclude that the Benders decomposition-based scheme based on the super-product master problem is superior to the benchmark methods with respect to the lower bound produced at termination.

Table 4: Average lower bound reported at termination

CBD	CPLEX	CPLEX-Benders	SPBD
1,272	66,232	42,449	98,971

Having established the effectiveness of using **EMP** in the context of Benders decomposition, we next assess the impact of the proposed valid inequalities. To do so, we compare the performance of **SPBD**, **SPBD1**, **SPBD2**, **SPBD3** and **SPBD123** with respect to the average optimality gaps and lower bounds reported at termination as well as the average number of feasibility and optimality cuts generated during execution. We present these results, averaged over all instances, in Table 5.

Table 5: Gaps, lower bounds and number of Benders cuts found by **SPBD**, **SPBD1**, **SPBD2**, **SPBD3** and **SPBD123**

Method	Opt. gap	Lower bound	Feasibility cuts	Optimality cuts
SPBD	44.34%	98,971	29.18	0.0
SPBD1	43.06%	101,201	15.82	0.0
SPBD2	12.65%	154,263	4.48	0.0
SPBD3	42.69%	101,863	9.67	0.0
SPBD123	8.86%	161,024	3.11	0.03

The use of each valid inequality leads to a decrease in the optimality gap at termination as well as an increase in the lower bound compared to **SPBD**. When considered individually, the first and third valid inequalities have a smaller impact than the second valid inequality. However, amongst these methods, the best results are obtained with **SPBD123**, indicating that all three valid inequalities, together, are the most effective. We also see that the valid inequalities have a significant impact on the number of feasibility cuts generated. Recalling that the valid inequalities are designed to render infeasible vehicle allocations that will not yield a feasible subproblem, this suggests the inequalities are having the intended effect. Ultimately, we conclude from this section that one reason the proposed algorithm produces solutions with a provably smaller optimality gap is that the new master problem and valid inequalities enable the algorithm to produce a much stronger lower bound than the benchmark methods.

5.3.3. Improving the upper bound

We next analyze the impact the heuristic has on the ability of **SPBD123H** to produce high-quality primal solutions. To do so, we measure for an instance and a method the improvement in the primal solution over that of the initial heuristic solution, (x_h, y_h), by computing the primal gap:

$$\text{primal-gap}_{UB}^{Method} = \frac{z(x_h, y_h) - \text{UB}_{Method}}{z(x_h, y_h)} \times 100$$

Here, UB_{Method} represents the objective function value of the best primal solution found by the method *Method* during its execution. We benchmark **SPBD123H** against **CPLEX** and **SPBD123**, and present averages of these primal gaps over instances with the same number of products in Table 6. Column "% *Method* impr." of Table 6 reports the percentage of instances for which the considered method was able to produce a primal solution with lower objective function value than the initial primal solution. The column "$\text{primal-gap}_{UB}^{Method}$" indicates the average primal gap obtained over instances for which the initial primal solution is improved.

Table 6: Comparison of upper bounds produced by **CPLEX**, **SPBD123** and **SPBD123H**

$\lvert\mathcal{P}\rvert$	% CPLEX impr.	$\text{primal-gap}_{UB}^{CPLEX}$	% SPBD123 impr.	$\text{primal-gap}_{UB}^{SPBD123}$	% SPBD123H impr.	$\text{primal-gap}_{UB}^{SPBD123H}$
100	70.00 %	4.71 %	6.67 %	4.17 %	100.00 %	12.48%
200	16.67 %	3.19 %	0.00 %	-	70.00 %	5.57%
300	16.67 %	2.26 %	0.00 %	-	46.67 %	1.76%
400	0.00 %	-	0.00 %	-	10.00 %	0.44%
500	0.00 %	-	0.00 %	-	0.00 %	-

We first observe that **CPLEX** has a better performance than **SPBD123** for every set of instances. Thus, implementing the **EMP** and the valid inequalities in the Benders strategy does not allow to obtain better primal solutions than those computed by **CPLEX**. However, with the addition of the slope-scaling heuristic into the Benders strategy, **SPBD123H** is able to improve upon the initial primal solution more often than **CPLEX**, and with a greater magnitude. Lastly, we note that as the number of products increases, all methods struggle to improve the initial primal solution.

We next directly compare the objective function values of the best primal solutions obtained by **SPBD123H** and those of the best solutions found by **CPLEX**. To do so, for each instance, we compute a primal gap:

$$\text{ub-gap} = \frac{\text{UB}_{CPLEX} - \text{UB}_{SPBD123H}}{\text{UB}_{CPLEX}} \times 100$$

In Table 7, we report the average primal gaps over instances with the same number of products. We see that

Table 7: Comparison of CPLEX, SPBD123H with respect to primal solution quality

$\lvert\mathcal{P}\rvert$	100	200	300	400	500
ub-gap	9.41%	3.36%	0.44%	0.04%	0%

SPBD123H outperforms **CPLEX** for sets of instances with 100 and 200 products. However, as both methods struggle to improve the upper bound for instances with larger numbers of products, the gap tends to 0 as the number of products grows.

We return our attention to Table 2, and the observation that the optimality gap reported by **SPBD123H** at termination decreases as the number of products increases. At the same time, we see that the impact of the heuristic on the ability of **SPBD123H** to produce improved primal solutions worsens as the number of products increases. We conclude from these observations that the ability of **SPBD123H** to produce strong lower bounds improves as the number of products increases. We (partially) attribute this to the fact that the size of the super-product master problem, **EMP**, (in terms of number of variables and constraints) is independent of $|\mathcal{P}|$. Thus, solving the master problem does not become more computationally challenging. At the same time, the number of valid inequalities does increase as the number of products increases. As the valid inequalities strengthen the master problem, more of them likely leads to a stronger lower bound. In addition, the demand volumes that must be routed in the master problem increase as the number of products increases. As these increased volumes likely require an increase in vehicle allocations, $y_{ij}^{tt'}$, we hypothesize that they also strengthen the master problem. Ultimately, we conclude from this section that one reason the proposed algorithm produces solutions with a provably smaller optimality gap is that the heuristic often enables the proposed algorithm to produce primal solutions with lower objective function values than the benchmark methods.

5.4. Results on a more general variant

To study the robustness of the proposed Benders-based approach, we next investigate its performance on the variant that considers all capacity constraints. Given the performance of the various methods on the first variant, we limit our comparison to **SPBD123H** and **CPLEX**. Specifically, we first compare the (average) optimality gaps reported by each method at termination. Like the first variant, we note that none of the instances could be solved by either method within the time limit. Note that, certain instances are infeasible when the capacity constraints (6) and (7) are in place. More specifically, one out of the 30 instances with 400 products becomes infeasible. Among instances with 500 products, 13 out of 30 become infeasible.

In Table 8, we report the optimality gaps obtained by **CPLEX** and **SPBD123H**. These gaps are averaged over instances with the same number of products $|\mathcal{P}|$.

Table 8: Optimality gaps comparison of **CPLEX** and **SPBD123H**

$\lvert\mathcal{P}\rvert$	CPLEX Opt. gap	SPBD123H Opt. gap
100	13.89%	5.72%
200	24.69%	7.95%
300	47.93%	6.87%
400	65.37%	4.63%
500	83.24%	3.08%

These results are quite similar to those corresponding to the first variant. For every set of instances our Benders strategy outperforms **CPLEX**. Again, the performance of **CPLEX** is significantly impacted by the number of products, while our Benders strategy remains effective for the largest instances.

As with the first variant, we next analyze the quality of the primal solutions produced by both **CPLEX** and **SPBD123H** over instances with the same number of products. In Table 9, we report the percentage of instances for which each method managed to improve the initial primal solution. We also display the average primal gap obtained over instances for which the initial primal solution is improved.

Table 9: Comparison of upper bounds produced by **CPLEX** and **SPBD123H**

$\lvert\mathcal{P}\rvert$	% CPLEX impr.	primal-gap$_{UB}^{CPLEX}$	% SPBD123H impr.	primal-gap$_{UB}^{SPBD123H}$
100	56.67%	2.75%	100.00%	12.09%
200	24.39%	0.79%	63.41%	5.98%
300	6.67%	0.18%	0.00%	-
400	0.00%	-	0.00%	-
500	0.00%	-	0.00%	-

We see that **SPBD123H** is again more likely than **CPLEX** to produce a primal solution that is better than the initial solution, and when it does, it is of higher quality.

6. Conclusions and future work

In this paper, we studied a transportation problem inspired by restaurant supply chains, the Logistics Service Network Design Problem (LSNDP). The goal of the LSNDP is to determine a cost-effective plan for transporting products from suppliers to customers through a multi-echelon distribution network. As these products are small relative to vehicle capacity, a critical strategy for achieving low transportation costs is consolidation. Specifically, to route products so that vehicles transport multiple products at a time, with each product potentially sourced by a different supplier and destined for a different customer.

As a result, the problem we studied can be viewed as a special case of the Service Network Design Problem (SNDP). However, the problem also has some special features, which the proposed algorithm exploits. For example, as suppliers feed (potentially multiple levels of) warehouses, which then feed customers, the LSNDP seeks to design a "forward flow" network. This is different from the type of network designed by the SNDP, wherein flows are typically omnidirectional. In addition, as customers request products that can be manufactured by multiple suppliers, the LSNDP also determines the supplier (and corresponding supply location in the network) for each customer request. This differs from the general SNDP typically studied in the literature, wherein the origins of shipments to be transported are known *a priori*. Relatedly, the instances we used to test the proposed algorithm are often much larger than the SNDP instances considered in the literature.

To solve the LSNDP, we proposed a Benders decomposition-based solution approach. More specifically, we proposed an algorithm based on the recently-proposed *Partial Benders Decomposition*, wherein information is retained in the master problem solved at an iteration in order to strengthen the bound it provides and speed up the convergence of the algorithm as a whole. Here, the information retained in the master problem is characterized by a "super-product" that is an aggregation of all the products to be routed. We proved the validity of this new master problem and computationally demonstrated the effectiveness of solving it in the context of a Benders decomposition-type algorithm instead of the master problem typically used for this type of problem. We proposed additional speed-up techniques, including valid inequalities and a heuristic for finding high-quality solutions. An extensive computational study illustrated that the resulting algorithm produced solutions that are of provably high-quality for different variants of the problem.

We see multiple avenues for future algorithmic work. In the context of solving the LSNDP, we intend to explore three potential enhancements to the method. The first enhancement is to develop Benders feasibility cuts customized for our problem. In the computational study, we observed that very few master solutions yield feasible subproblems. Thus, strenghtening the feasibility cuts can significantly speed up the convergence of the algorithm. A related opportunity is to derive combinatorial Benders cuts [4] from infeasible subproblems. The second enhancement is to consider multiple super-products in the master. In that case, the number of super-products should be an input parameter. This parameter would determine how many sets to partition the set of products into, with a super-product created for each product set. The third enhancement would again consider multiple super-products, but in a dynamic manner. Namely, the number of super-products considered in the master problem would change during the course of the Benders algorithm. More generally, while we have focused our algorithmic work on the LSNDP as it is the problem faced by our industrial partner, many of the proposed algorithmic ideas are also valid for the general SNDP. Thus, another avenue for future work is to adapt our approach to the general SNDP, or to SNDP variants that address practical features such as the management of the vehicle fleet or the consideration of products with heterogeneous sizes.

Acknowledgements

This work has been partially supported by the French National Research Agency through the Pi-Comodalité project under the grant ANR-15-CE22-0012. This support is gratefully acknowledged.

References

[1] Beamon, B. M. (1998). Supply chain design and analysis::models and methods. *International journal of production economics*, 55(3):281–294.

[2] Benders, J. F. (1962). Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252.

[3] Cheong, M. L., Bhatnagar, R., and Graves, S. C. (2007). Logistics network design with supplier consolidation hubs and multiple shipment options. *Journal of Industrial and Management Optimization*, 3(1):51.

[4] Codato, G. and Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766.

[5] Cordeau, J.-F., Pasin, F., and Solomon, M. M. (2006). An integrated model for logistics network design. *Annals of operations research*, 144(1):59–82.

[6] Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, 32:1429–1450.

[7] Costa, A. M., Cordeau, J. F., Gendron, B., and Laporte, G. (2012). Accelerating benders decomposition with heuristic master problem solutions. *Pesquisa Operacional*, 32:03–20.

[8] Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122:272–288.

[9] Crainic, T. G., Gendron, B., and Hernu, G. (2004). A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10:525–545.

[10] Crainic, T. G., Hewitt, M., and Rei, W. (2014a). Partial decomposition strategies for two-stage stochastic integer programs. *CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise.*

[11] Crainic, T. G., Hewitt, M., Toulouse, M., and Vu, D. M. (2014b). Service network design with resource constraints. *Transportation Science*, 50:1380–1393.

[12] Crainic, T. G., Rei, W., Hewitt, M., and Maggioni, F. (2016). Partial benders decomposition strategies for two-stage stochastic integer programs. *CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise* .

[13] Dufour, É., Laporte, G., Paquette, J., and Rancourt, M.-È. (2018). Logistics service network design for humanitarian response in east africa. *Omega*, 74:1–14.

[14] Fontaine, P., Crainic, T., Rei, W., and Jabali, O. (2017). *Multi-modal scheduled service network design with resource management for two-tier city logistics*. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise.

[15] Magnanti, T. L. and Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29:464–484.

[16] Melo, M. T., Nickel, S., and Saldanha-Da-Gama, F. (2009). Facility location and supply chain management–a review. *European journal of operational research*, 196(2):401–412.

[17] Pedersen, M. B., Crainic, T. G., and Madsen, O. B. (2009). Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43:158–177.

[18] Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259:801–817.

[19] Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2018). Accelerating the benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization*, 28:875–903.

[20] Rei, W., Cordeau, J.-F., Gendreau, M., and Soriano, P. (2009). Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2):333–345.

[21] Srivastava, S. K. (2008). Network design for reverse logistics. *Omega*, 36(4):535–548.

[22] Wang, Z., Qi, M., Cheng, C., and Zhang, C. (2019). A hybrid algorithm for large-scale service network design considering a heterogeneous fleet. *European Journal of Operational Research*.

[23] Wieberneit, N. (2008). Service network design for freight transportation: a review. *OR Spectrum*, 30:77–112.

[24] Zetina, A. C., Cordeau, J.-F., and Contreras, I. (2017). *Exact algorithms for the multicommodity uncapacitated fixed-charge network design problem*. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise.

Appendix A.

Theorem 1. *The enhanced master problem,* **EMP***, is a relaxation of the Logistics Service Network Design problem, LSND($\mathcal{G}_\mathcal{T}$).*

Proof. We prove this claim by showing that any feasible solution to the LSND($\mathcal{G}_\mathcal{T}$) is also feasible for the **EMP** and has the same objective function value. To do so, we let (x, y) be a feasible solution of the LSND($\mathcal{G}_\mathcal{T}$). Consider a solution (x^χ, y, z) such that:

$$x_{ij}^{\chi tt'} = \sum_{p\in\mathcal{P}} x_{ij}^{ptt'},\ \forall((i,t),(j,t')) \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, \qquad z = \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} c_{ij} x_{ij}^{\chi tt'} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_\mathcal{T}} c_{ii} x_{ii}^{\chi tt'}$$

It is easy to prove this solution is feasible for the enhanced master problem. By construction, for each variable $x_{ij}^{ptt'}$ in the LSNDP, there is a corresponding variable $x_{ij}^{\chi tt'}$ in **EMP**. We know that for each warehouse $(j, t') \in \mathcal{W}_\mathcal{T}$ and each product $p \in \mathcal{P}$: $\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{ij}^{ptt'} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{jl}^{pt't''} = 0$. If we sum this expression on the products, we obtain:

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} \sum_{p\in\mathcal{P}} x_{ij}^{ptt'} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} \sum_{p\in\mathcal{P}} x_{jl}^{pt't''} = 0 \iff \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{ij}^{\chi tt'} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_\mathcal{T}\cup\mathcal{H}_\mathcal{T}} x_{jl}^{\chi t't''} = 0$$

Therefore, (x^χ, y, z) respects constraints (19). As (x, y) respects constraints (3)-(4)-(5)-(6)-(7), it is trivial to demonstrate (x^χ, y, z) also respects constraints (20)-(21)-(22)-(23)-(7). By construction of z, (x^χ, y, z) respects constraint (24) which makes it an admissible solution to the enhanced master problem.

Let $Q(x, y)$ be the objective value of (x, y):

$$Q(x,y) = \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} f_{ij} y_{ij}^{tt'} + \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} \sum_{p\in\mathcal{P}} c_{ij} x_{ij}^{ptt'} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_\mathcal{T}} \sum_{p\in\mathcal{P}} c_{ii} x_{ii}^{ptt'}$$

$$= \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} f_{ij} y_{ij}^{tt'} + \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} c_{ij} x_{ij}^{\chi tt'} + \sum_{((i,t),(i,t+1))\in\mathcal{H}_\mathcal{T}} c_{ii} x_{ii}^{\chi tt'} = \sum_{((i,t),(j,t'))\in\mathcal{A}_\mathcal{T}} f_{ij} y_{ij}^{tt'} + z = Q(x^\chi, y, z)$$

Solution (x^χ, y, z) that replicates solution (x, y) by an aggregation of flows, is feasible for the enhanced problem. The two solutions have identical objective function value. Thus EMP is a relaxation of the LSNDP. □

Theorem 2. *The proposed inequalities (28)-(29) are valid.*

Proof. We first define the LSNDP over a time-expanded graph with super-sources. We demonstrate that the new problem is equivalent to the original one. Then, we demonstrate that any feasible solution for the new problem is equivalent to a solution for the **EMP** that respects (28)-(29).

Let (x, y) be a feasible solution for the LSND($\mathcal{G}_\mathcal{T}$). Let us extend $\mathcal{G}_\mathcal{T}$ similarly to what was done in subsection 4.2.1. For each $p \in \mathcal{P}$, we add a super-source ss_p to $\mathcal{G}_\mathcal{T}$. In addition, for each $(s, t) \in \mathcal{S}_\mathcal{T}$ such that $p \in \mathcal{P}^s$, we add a time-expanded arc $((ss_p), (s, t))$ with null linear cost and null fixed cost to $\mathcal{G}_\mathcal{T}$. We name the new time-expanded network as $\mathcal{G}_{\mathcal{T}}^{+}$. On each time-expanded arc $((ss_p),(s,t)) \in \mathcal{G}_{\mathcal{T}}^{+}$, let us define a continuous variable $x_{ss_p}^{pt}$. Let us add the following constraints to the LSND($\mathcal{G}_{\mathcal{T}}^{+}$):

$$\sum_{((ss_p),(j,t))\in\mathcal{A}_{\mathcal{T}}} x_{ss_pj}^{pt} \geq D^p, \qquad \forall p \in \mathcal{P} \tag{A.1}$$

$$\sum_{((i,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} x_{ij}^{ptt'} - \sum_{((j,t'),(l,t''))\in\mathcal{A}_{\mathcal{T}}} x_{jl}^{pt't''} = 0, \qquad \forall (j,t') \in \mathcal{S}_{\mathcal{T}} \tag{A.2}$$

We now extend a solution for the LSND($\mathcal{G}_{\mathcal{T}}$) to a solution for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$). By construction, for each $(s,t) \in \mathcal{S}_{\mathcal{T}}$ and for each $p \in \mathcal{P}^s$, the only arc of $\mathcal{G}_{\mathcal{T}}^{+}$ incoming to (s,t) such that a flow variable is defined for product p is $((ss_p),(s,t))$. Thus, for each $(s,t) \in \mathcal{S}_{\mathcal{T}}$, the only way to satisfy constraints (A.2) is to set the flow value of product p on arc $((ss_p),(s,t))$ to $\sum_{((s,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} x_{sj}^{ptt'}$. As the original solution satisfies all customer demands, for each $p \in \mathcal{P}$ we have $\sum_{(s,t)\in\mathcal{S}_{\mathcal{T}}} \sum_{((s,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} x_{sj}^{ptt'} \geq D^p$. Thus, the extended solution satisfies constraints (A.1) and is feasible for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$).

Each solution for the LSND($\mathcal{G}_{\mathcal{T}}$) admits a single corresponding solution for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$). In addition, both solutions have identical objective values. Thus, the LSND($\mathcal{G}_{\mathcal{T}}$) is equivalent to the LSND($\mathcal{G}_{\mathcal{T}}^{+}$).

Let $(x,y)^{+}$ be a feasible solution for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$). Let $(x^{\mathcal{X}},y,z)$ be the solution for the **EMP** that replicates $(x,y)^{+}$ by an aggregation of flows. As $(x,y)^{+}$ respects constraints (A.1) and (A.2), by construction $(x^{\mathcal{X}},y,z)$ respects constraints (28) and (29). Thus, constraints (28) and (29) do not cut off $(x^{\mathcal{X}},y,z)$ that replicates a feasible solution for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$), and inequalities (28) and (29) are valid. □

Theorem 3. *The proposed inequality (30) is valid.*

Proof. Let (x,y) be an optimal solution for the LSND($\mathcal{G}_{\mathcal{T}}$). Let $((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}$ such that $(i,t) \in \mathcal{S}_{\mathcal{T}}$ and $(j,t') \in \mathcal{C}_{\mathcal{T}}$. For each product $p \in \mathcal{P}^i$, $x_{ij}^{ptt'}$ cannot be greater than $d_{jt'}^{p}$, as otherwise (x,y) would not be optimal for the LSND($\mathcal{G}_{\mathcal{T}}$). As a result, $\sum_{p\in\mathcal{P}^i} x_{ij}^{ptt'} \leq \sum_{p\in\mathcal{P}^i} d_{jt'}^{p}$. Let $(x^{\mathcal{X}},y,z)$ be a solution for the **EMP** solution that replicates (x,y) by an aggregation of flows. By construction, for each $((i,t),(j,t')) \in \mathcal{A}_{\mathcal{T}}$ such that $(i,t) \in \mathcal{S}_{\mathcal{T}}$ and $(j,t') \in \mathcal{C}_{\mathcal{T}}$, we have $x_{ij}^{\mathcal{X}tt'} = \sum_{p\in\mathcal{P}^i} x_{ij}^{ptt'} \leq \sum_{p\in\mathcal{P}^i} d_{jt'}^{p}$. Thus, constraints (30) do not cut off $(x^{\mathcal{X}},y,z)$ that replicates an optimal solution for the LSNDP, and inequality (30) is valid. □

Theorem 4. *The proposed inequality (31) is valid.*

Proof. Let (x,y) be a feasible solution for the LSND($\mathcal{G}_{\mathcal{T}}$). Let consider $p \in \mathcal{P}$ and $t^* \in \mathcal{T}$ such that $\bar{D}_{t^*}^{p} > \bar{D}_{t^*-1}^{p}$. Thus, there exists a customer $(c,t^*) \in \mathcal{C}_{\mathcal{T}}$ such that $d_{ct^*}^{p} > 0$. $t_{ss_p}^{min}$ is the smallest transit time between all supplier of product p and a customer for its product. Thus, the total amount of product p shipped from suppliers before or at time $t^* - t_{ss_p}^{min}$ must be greater or equal than $\bar{D}_{t^*}^{p}$, i.e.:

$$\sum_{\substack{(s,t)\in\mathcal{S}_{\mathcal{T}} \\ t\leq t^*-t_{ss_p}^{min}}} \sum_{((s,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} x_{sj}^{ptt'} \geq \bar{D}_{t^*}^{p}$$

As in Theorem 2, we extend (x, y) and obtain a feasible solution $(x, y)^+$ for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$). By construction, we have:

$$\sum_{\substack{((ss_p),(s,t))\in\mathcal{A}_{\mathcal{T}}\\ t\leq t^*-t_{ss_p}^{min}}} {x_{ss_ps}^{pt}}^{+} = \sum_{\substack{(s,t)\in\mathcal{S}_{\mathcal{T}}\\ t\leq t^*-t_{ss_p}^{min}}} \sum_{((s,t),(j,t'))\in\mathcal{A}_{\mathcal{T}}} x_{sj}^{ptt'} \geq \bar{D}_{t^*}^{p}$$

Let $(x^{\mathcal{X}}, y, z)$ be the **EMP** solution that replicates $(x, y)^+$ by an aggregation of flows. By construction, we have:

$$\sum_{\substack{((ss_p),(s,t))\in\mathcal{A}_{\mathcal{T}}\\ t\leq t^*-t_{ss_p}^{min}}} x_{ss_ps}^{\mathcal{X}t} = \sum_{\substack{((ss_p),(s,t))\in\mathcal{A}_{\mathcal{T}}\\ t\leq t^*-t_{ss_p}^{min}}} {x_{ss_ps}^{pt}}^{+} \geq \bar{D}_{t^*}^{p}$$

Thus, constraints (31) do not cut off $(x^{\mathcal{X}}, y, z)$ that replicates a feasible solution for the LSND($\mathcal{G}_{\mathcal{T}}^{+}$), and inequality (31) is valid. □

Appendix B.

Table B.1: Comparison of the lower/upper bounds produced by **CBD**, **CPLEX-Benders**, **CPLEX** and **SPBD123H**

	CBD		CPLEX-Benders		CPLEX		SPBD123H	
$\lvert\mathcal{P}\rvert$	LB	UB	LB	UB	LB	UB	LB	UB
100	1,022	67,214	15,481	67,214	56,329	64,983	55,858	58,728
200	1,301	124,216	30,193	124,216	72,915	123,577	111,225	119,353
300	1,413	173,951	43,475	173,951	69,271	173,324	160,953	172,498
400	1,515	225,885	59,621	225,885	68,969	225,885	212,999	225,787
500	1,107	275,959	63,672	275,959	63,672	275,959	263,951	275,959